



---

# SHOPASSIST 2.0

---

GenAI based laptop shopping assistant



# ShopAssist Chatbot Project Report

## 1. Project Overview

Project Name: ShopAssist Chatbot for Laptop Recommendations

### Objective:

To create a chatbot that recommends laptops based on user requirements by leveraging a dataset containing laptop specifications, product names, descriptions, and other relevant details.

### Problem Statement:

Given a dataset containing information about laptops (such as product names, specifications, descriptions), build a chatbot that can parse this data and provide accurate laptop recommendations aligned with user requirements.

### Solution Summary

ShopAssist 2.0 is designed to interact with users conversationally, asking relevant questions to capture their needs and preferences for a laptop. Using advanced language model capabilities, it provides tailored recommendations, narrowing the selection to the **top 3** laptops best matching the user's profile.

### Approach

The development of ShopAssist can be broken down into several stages:

## 1. Conversation and Information Gathering:

- Goal: Engage the user to understand their requirements, such as display quality, processing speed, GPU intensity, portability, multitasking capability, and budget.
- Implementation: The chatbot uses natural language processing (NLP) to manage conversational flow, asking targeted questions to gather detailed specifications from the user.

## 2. Information Extraction:

- Goal: Transform user inputs into structured data that can be used to filter and score laptops from the dataset.
- Implementation: Once sufficient information is gathered, rule-based functions are used to extract key data points and rank laptops based on compatibility with the user's requirements.

## 3. Personalized Recommendation:

- Goal: Provide a shortlist of suitable laptops and continue the conversation to refine recommendations based on follow-up user questions.
- Implementation: Using the extracted information, ShopAssist presents a list of up to 3 laptops that align with the user's preferences and further engages with the user to address any specific concerns or questions about the recommendations.

## Technical Design

The chatbot's technical design involves a combination of APIs, function calling, and custom functions to meet the requirements. Below diagram is an outline of the core components:

### *Language Model Integration:*

- **Platform:** OpenAI language models power the conversational and recommendation abilities, utilizing GPT for generating responses.
- **Function Calling API:** Function calls help capture structured user preferences into a "user requirements dictionary."

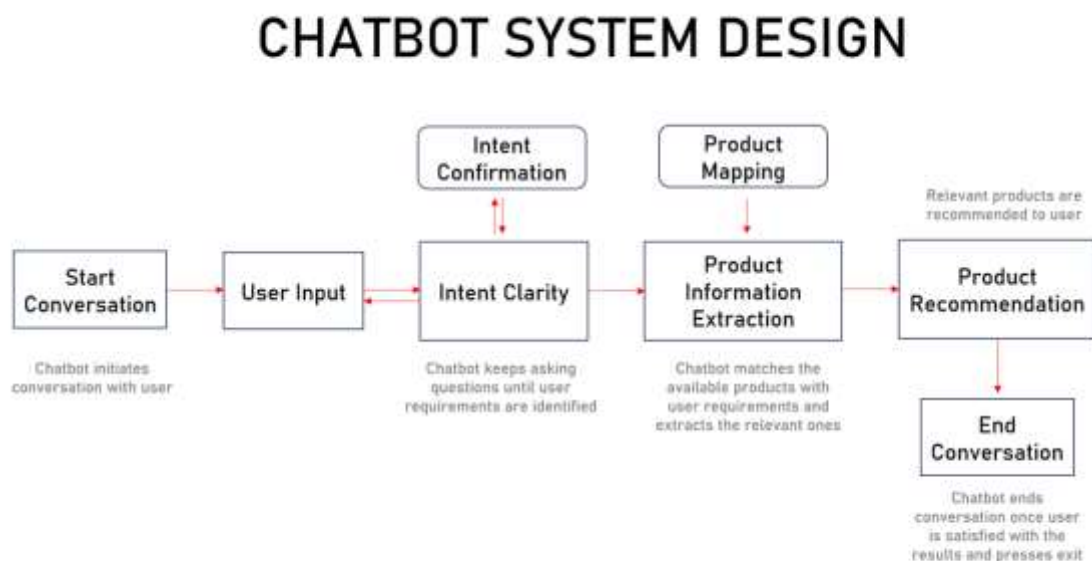
## Dialogue Management System:

### Overview:

A loop-based dialogue system guides the conversation, handles moderation checks, and ensures responses align with the user's intent.

### Code Details:

- The `dialogue_mgmt_system` function initializes the conversation and continues until either a suitable recommendation is found or the conversation is terminated.
- Moderation Checks: Each response is verified through `moderation_check` to filter inappropriate content.
- User Intent Confirmation: A confirmation layer validates the user's needs before proceeding with recommendations.
- Data Extraction: The function `get_chat_completions_with_function_calling` creates a structured dictionary containing user specifications by calling custom functions, which map key features like GPU intensity, display quality, and budget.



### Laptop Recommendation Process:

- Comparison and Filtering: The chatbot compares user preferences with the available laptops in the dataset using `compare_laptops_with_user`.
- Validation: Validates the top 3 recommendations through `recommendation_validation` to ensure they closely match user criteria.

- Final Interaction: The chatbot confirms the final recommendations through an additional conversation round, allowing the user to make the final decision or restart the process if requirements change.

## Key Functions and Components

1. ``get_chat_completions_with_function_calling(input, include_budget)`` :

- This function collects user input, invokes OpenAI's function calling API, and returns a structured dictionary representing the user's laptop specifications, including attributes like GPU intensity, display quality, and budget if applicable.

2. ``extract_user_profile_info`` :

- Purpose: Custom function to extract and structure user information based on inputs.
- Parameters: Receives specific attributes (e.g., GPU intensity, display quality, portability) and outputs a dictionary for further processing.

## Results and Insights

Through a structured conversation flow and precise information extraction, ShopAssist 2.0 effectively narrows down the available options and recommends laptops based on specific user needs. Testing indicates that ShopAssist 2.0 can provide personalized recommendations with high accuracy, improving user experience and satisfaction.